AD-A104 339 BEDFORD RESEARCH ASSOCIATES MA F/G 9/2
OCLC TAPE CONVERSION PROJECT,(U)
JUN 80 E CRONIN, S NEVINS F19628-79-C-0163
UNCLASSIFIED SCIENTIFIC-3 AFGL-TR-81-0055 NL

1 OF 1
AD
A104339

END
DATE FILED
10-81
DTIC

AFGL-TR-81-0055

AD A104339

OCLC TAPE CONVERSION PROJECT

E. Cronin
S. Nevins

Bedford Research Associates
2 DeAngelo Drive
Bedford, Massachusetts 01730

Scientific Report No. 3

30 June, 1980

Approved for public release; distribution unlimited

AIR FORCE GEOPHYSICS LABORATORY
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
HANSCOM AFB, MASSACHUSETTS 01731

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFGL-TR-81-0055	2. GOVT ACCESSION NO.: AD-A104 939
3. RECIPIENT'S CATALOG NUMBER:	
4. TITLE (and Subtitle): OCLC TAPE CONVERSION PROJECT	5. TYPE OF REPORT & PERIOD COVERED: Scientific Report No. 3
	6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): E. Cronin* S. Nevins	8. CONTRACT OR GRANT NUMBER(s): F19628-79-C-0163
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Bedford Research Associates, 2 DeAngelo Drive, Bedford, MA. 01730	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62101F, 9993XXXX
11. CONTROLLING OFFICE NAME AND ADDRESS: Air Force Geophysics Laboratory, Hanscom AFB, Massachusetts 01731, Monitor/Paul Tsipouras/SUWA	12. REPORT DATE: 30 June 1980
	13. NUMBER OF PAGES: 54
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

* Air Force Geophysics Laboratories
Analysis and Simulation Branch, (SUWA)
ESD Hanscom AFB, MA. 01731

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

OCLC TAPE, MODIFIED MARC FORMAT, CODE CONVERSION.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report explains the design and the use of a program to convert 9 track OCLC tapes (in modified MARC format) to 7 track tapes in a 6-bit code.

DD FORM 1 JAN 73 1473

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

INTRODUCTION

This paper explains the design and use of a program to convert 9-track OCLC tapes (in modified MARC format) to 7-track tapes. The 9-track tapes are written in an 8-bit expanded ASCII character code, and the 7-track tapes are written in a 6-bit code derived from the 8-bit code. These codes are unrecognizable by the machine; hence the need for this program.

(For an explanation of the MARC format for tapes, see reference (1)).

A few definitions are in order before proceeding:

bit - 1 binary digit. Each word in the machine consists of 60 bits. Numbering of the bits is inconsistent: for the AFGL utilities, numbering is 1-60, from right to left in the word; in all other instances, numbering is 0-59, from right to left.

byte - a character. In the 6-bit code, it is one group of 6-bits. In the 8-bit code, it is one 8-bit group. Bytes in a word are numbered 1-10 from left to right.

physical record - a block of information on the tape. For this program, physical records can be any size up to 2058 bytes. A physical record will never contain more than one logical record.

logical record - one complete record of information (leader, record directory, and all related control/variable fields). Logical records can be any size. If a logical record is greater than the maximum size for a physical record, it will spill over onto as many physical records as required.

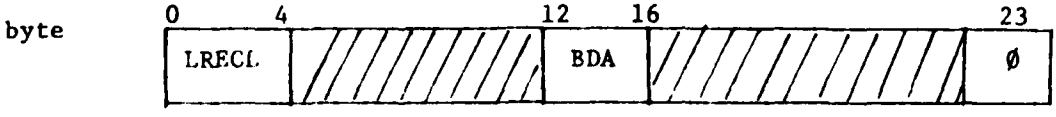
RECORD STRUCTURE

LEADER	RECORD DIRECTORY	CONTROL FIELDS	VAIRABLE FIELDS

All logical records on the 7 and 9-track tapes are in the format shown above. The length of a logical record is unrestricted. Physical records, however, will never be greater than 2048 bytes. If a logical record is greater than 2048 bytes, it spills over onto as many physical records as necessary. During the conversion, the leader and record directory are removed from the record to permit updating of certain fields. After the control and variable fields are converted, the leader and record directory are returned to their appropriate locations.

LEADER

The leader is fixed at 24 bytes. There are 2 fields which are modified during the conversion: the logical record length (LRECL) and the base data address (BDA). Also it has been found that some of the records on the 9-track tape contain information in byte 23 which, when converted, would change the length of the leader to 25 bytes. Therefore, byte 23 is set to Ø regardless of what is on the 9-track tape. This is consistent with the MARC format.



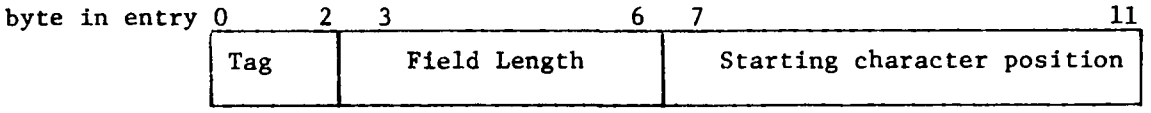
LRECL is the total number of bytes in the logical record. The number is right-justified with leading zeros.

BDA is the address at which the control fields begin in the record. Hence, it is equal to the sum of the length of the leader and the length of the record directory (including the record directory field terminator). The number is right justified with leading zeros.

RECORD DIRECTORY

The record directory is made up of a series of fixed length entries (12 bytes each) which contain the identification tag, the length, and the starting character position in the record of each control/variable field. The starting character position is relative to the BDA. Referring to the sample record in Appendix A, although the first control field begins at byte 145 of the record, the starting character position for the field is ØØØØØ. (145 is the BDA).

SAMPLE ENTRY

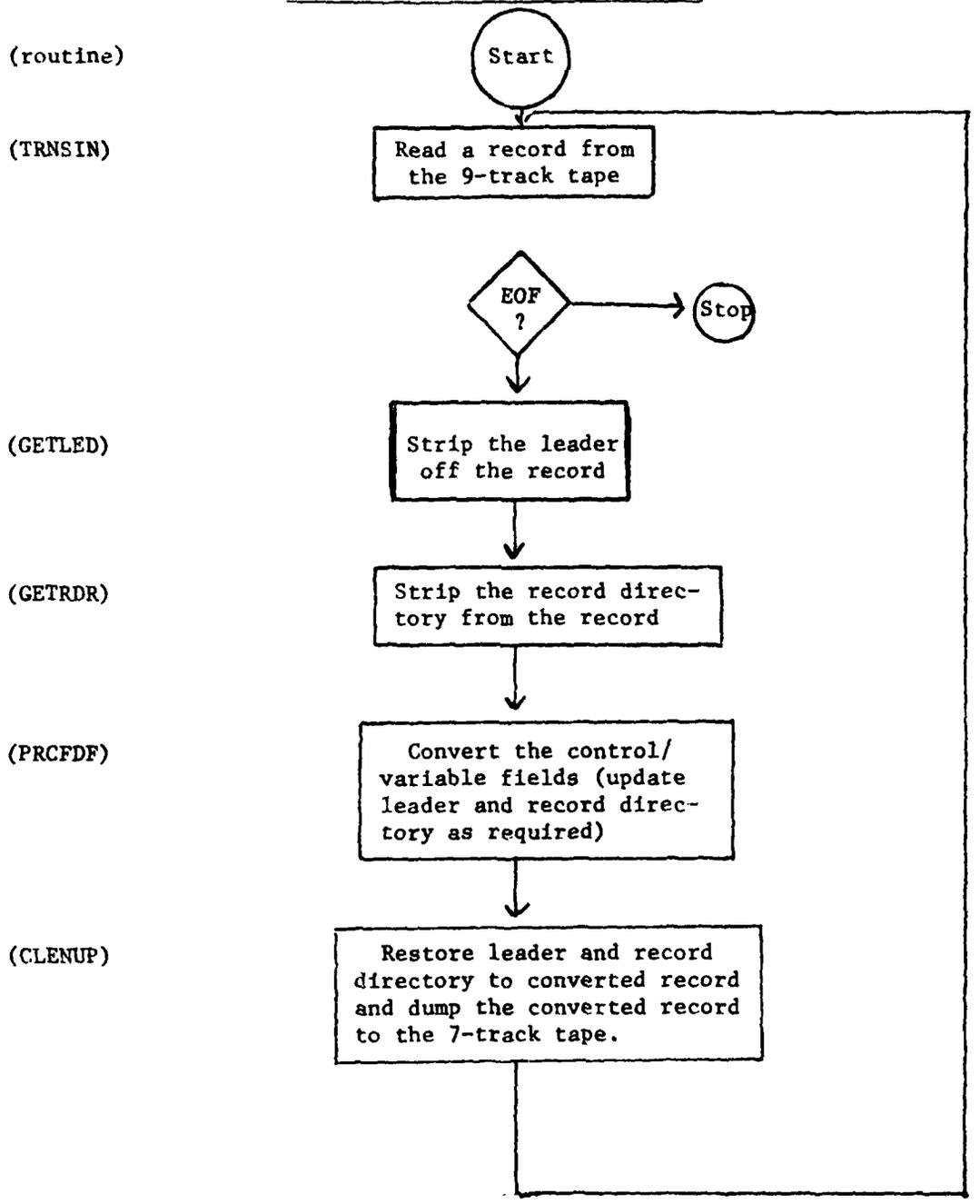


The end of the record directory is indicated by a field terminator ($1E_{16}$, 8-bit; 7336_8, 6-bit).

CONTROL/VARIABLE FIELDS

The control/variable fields do not, as of this writing, contain any information which needs to be updated (pointers, counters, etc.). They are converted verbatum, one field at a time. As a field is processed, the corresponding entry in the record directory is updated as required. Fields are separated by field terminators ($1E_{16}$, 8-bit; 7336_8, 6-bit), except for the last field in the record, which is followed by a record terminator ($1D_{16}$, 8-bit; 7335_8, 6-bit).

PROCESS OUTLINE



CONVERSION PROCESS

Each logical record is read directly into a buffer (no conversion is performed by the system). Since the information is on 8-bit bytes, and the machine operates on 6-bit bytes, bit manipulation is required to extract the bytes from the buffer.

A 60-bit word (CDC standard word size) will contain 7 8-bit bytes, plus a half byte (4 bits). Some information will, therefore, span word boundaries. For example, word Ø (the first word in a record) will contain bytes Ø-6, plus the first 4 bits of byte 7. Word 1 will contain the last 4 bits of byte 7, and bytes 8-14. Word 2 contains bytes 15-21, and the first 4 bits of byte 22, etc.

The information is extracted from the buffer 1 byte at a time, and converted to its 6-bit equivalent by the following method:

> A translation table has been created in the form of a vector with 256 elements. The 8-bit codes can have values in the range $0\text{-}255_{10}$. As each 8-bit code is extracted, it is stored right-justified in an integer word and incremented by 1. This word is then used as an index to the translation vector. That element in the vector contains the 6-bit equivalent for the 8-bit code. For example, (using Appendix A and B), the 8-bit code for a semi-colon (;) is $3B_{16}$, or $59_{1Ø}$. This implies that entry $6Ø_{1Ø}$ in the translation vector contains 33_8, which is the 6-bit code for a semi-colon. Likewise, the 8-bit code for a line-feed is $ØA_{16}$, or $1Ø_{1Ø}$. Entry $11_{1Ø}$, therefore, contains 7312_8.

Some entries in the table contain 7777_8. This indicates that the code is not presently in use (no cross reference between the 8 and 6-bit codes exists). The translation table can be easily modified should these codes become active. If the program encounters any of these codes, a message is printed indicating the undefined 8-bit code. When these codes are added to the output buffer, they appear as 37_8 (a question mark).

SAMPLE JOB

This sample job assumes that the source code has been compiled, and that the object code exists in a permanent file NTOSX3465. The 9-track input tape is labeled XØ3963, and is approached as a stranger tape via Unit 1. The 7-track output tape is labeled CC3387 accessed via Unit 2.

Job Card.

```
VSN(TAPE1=XØ3963/NT,TAPE2=CC3387).
REQUEST,TAPE1,NT,5,E.   (XØ3963/NORING)
PAUSE.  OKAY TO WRITE ON DECHICHIO TAPE
REQUEST,TAPE2,S,N,HI,RING.   (CC3387/RING/DECHICHIO)
ATTACH,NTOS,NTOSX3465,ID=NEVINS.
NTOS.
(end of job)
```

The program requires no input, other than the tape.

OUTPUT

The program will indicate any codes encountered on the input tape which do not have a 6-bit equivalent. A self-explanatory message is printed, and the code indicated.

If a problem occurs with either the input or output tape, a message is printed indicating the problem. A message of this type is due to either a bad tape or tape drive.

Although it is unlikely, the user may receive a message indicating that buffer sizes have become inadequate. This is a minor problem, but it does require a few changes in the program. Should the situation occur, the message will indicate the appropriate action to be taken.

Upon successful completion of the task, a message is printed.

INTERNAL ROUTINES

(subroutine/Function names preceded by * are AFGL utility routines)

ADDLED (no agruments)

Commons: LEAD, OUT Subroutines/Functions: INTCHR, *MXGETX, *MSPUTX

Adds the leader to the beginning of the output buffer. The logical record length and base data adress are first converted to character code, then the entire leader is added.

ADDRDR (no agrments)

Commons: OUT, DIREC Subroutines/Functions: INTCHR, *MXGETX, *MXPUTX

Adds the record directory to the output buffer immediately following the leader. The field length and starting address for each entry in the directory are converted to character code, and the entire entry is then inserted in the buffer. A field terminator (7336_8) is inserted after the last entry.

BLOCK DATA

Common

TRANS contains the translation table for the 8 to 6 bit conversion. The numeric value for the 8-bit code corresponds to the location in the table where the 6-bit code resides. For 8-bit codes which do not have a 6-bit equivalent, the location contains 7777_8.

BUFR contains the input buffer, number of words in the buffer, number of bits in the last word, end-of-file flag, word being processed, next byte to be processed, and next bit to be processed.

DIREC contains the record directory, and the number of entries in the directory. Each row in the array corresponds to independent entries in the directory, and the columns contain:

Column 1 - tag (character code)
Column 2 - field length (integer)
Column 3 - starting address (integer)

LEAD contains the leader (packed in 2.4 words, character code), the logical record length (LRECL, integer), and the base data address (BDA, integer).

OUT contains the output buffer (space for 3205 word records), the number of words used in each buffer section, the section of the buffer currently being used, and the next available byte in the buffer.

BRKDIN (INT, 11, 110, 1100, ITH, IHT)

Commons: None Subroutines/Functions: None

Breaks the integer INT down into its component digits. Maximum number which can be broken down is 99,999. If INT is greater than that, IHT will have a value greater than 9.

CHRINT (CHR, NCHR, INT)

Commons: None Subroutines/Functions: *MXGETX

Converts the NCHR characters (right-justified) in CHR from the 6-bit character code to an integer. The result returns through INT.

CLENUP (no agruments)

Commons: OUT Subroutines/Functions: ADDLED, ADDRDR, TRNSOT

Cleans up the output buffer after the entire record has been processed. The leader and record directory are inserted in the buffer at the appropriate locations, and the buffer is dumped to the 7-track tape in 205 words (2048 byte) segments.

CONVRT (ATEBIT,SIXBIT)

Commons: TRANS Subroutines/Functions: None

Converts the 8-bit code in ATEBIT to the 6-bit equivalent and returns it through SIXBIT. ATEBIT is incremented by 1 and used as the index to the array containing the translation talbe. The location pointed to will contain the 6-bit equivalent. SIXBIT will receive either 1 or 2 bytes right-justified.

GENOUT (TWD)

Commons: LEAD Subroutines/Functions: *MXGETX, PUTBT

Determines whether 1 or 2 bytes are to be added to the buffer from TWD. The information is added a byte at a time, from left to right, using PUTBT.

GETLED (no arguments)

Commons: BUFR, LEAD Subroutines/Functions: CHRINT, CONVRT, GET8BT, *MXPUTX

Strips the leader off the input buffer. Each byte is converted as it is pulled off. Byte 23 is forced to a Ø (character code) because the OCLC tapes have some records with information in that location which, when converted, would make the leader 25 bytes long. The programs which use the tape do not expect information in that field, hence the Ø.

The logical record length (LRECL) and base data address (BDA) are pulled off the leader and converted to integers.

GETRDR (no arguments)

Commons: BUFR, DIREC, LEAD Subroutines/Functions: CHRINT, CONVRT, GET8BT, *MXPUTX

Strips the record directory from the input buffer. Information is taken from the buffer 12 bytes at a time until a field terminator is encountered. Each 12 byte group contains one entry in the directory. Each entry contains a tag, field length, and starting address. The field length and starting address are converted to integer.

The LRECL and BDA are incremented, because the field terminator at the end of the directory becomes 2 bytes after conversion.

GET8BT (ATEBIT)

Commons: BUFR Subroutines/Functions: *MXIFTX, XTRACT

Extracts the next 8 bits from the input buffer, and returns them through ATEBIT.

INTCHR (INT, NDIG, CHR)

Commons: None Subroutines/Functions: BRKDIN, *MXPUTX

Converts NDIG digits from the integer INT to the 6-bit character code. The integer is broken down 1 digit per word. Each word is then OR'd with 20_8. wjoc tje 6-bit character code for that digit. The codes are then placed right-justified into CHR.

MAIN (entry routine)

Commons: BUFR Subroutines/Functions: CLENUP, GETLED, GETRDR, PRCFDF, TRNSIN

This is the entry routine for that program. TRNSIN, GETLED, GETRDR, PRCFDF, and CLENUP are called in that order to process each record. The loop continues until the end-of-file flag in BUFR is turned on by TRNSIN, at which point the program stops.

OUTINT (no agrments)

Commons: LEAD, OUT Subroutines/Functions: None

Initializes the output common. The buffer is cleared to zero, and the BDA is used to determine the position in the output buffer where the control/variable fields can start.

PRCFDF

Commons: DIREC Subroutines/Functions: CONVRT, GENOUT, GET8BT, OUTINT

Translates the control/variable fields from the input buffer. The field length and starting address for each field is computed and stored in the record directory immediately after the field is processed. Processing continues until a record terminator is encountered.

PUTBT(TWD)

Commons: OUT Subroutines/Functions: *MXPUTX

The rightmost byte from TWD is added to the output buffer at the next available position.

TRNSIN (no arguments)

Commons: BUFR Subroutines/Functions: LENGTHX, UNIT

Retrieves the next record from the 9-track tape. The length of the record is determined and stored in BUFR. If an end-of-file occurs, a flag in BUFR is turned on.

TRNSOT (OUTP, STWD, ENWD)

Commons: None Subroutines/Functions: UNIT

Write information from array OUTP, from word, STWD to word ENWD, then to the 7-track tape.

XTRACT (WORD1, NBIT, WORD2, STBIT)

Commons: None Subroutines/Functions: *ISBYTX, *MX1FTX

Extracts NBIT bits from the left of WORD1, and inserts them in WORD2 at STBIT. WORD1 is shifted left circular NBIT bits.

EXTERNAL ROUTINES

AFGL Utility Routines

ISBYTX (I,N,M,J) Bit string function

Picks N bits from the right of word J and inserts them starting at bit I of word M. Negative N appears as left shift (I-1). Negative I picks from the opposite end of the word.

MXIFTX (J,I) Shifting function

If I is positve, the 60 bit word J is shifted left circular I bit positions. If I is negative, J is shifted right with the sign bit filling the left side of the word, and the bits shifted off the right of the word are lost.

MXGETX (A,I,N) Pick Bytes (function)

Pick N bytes, starting at byte I, from A and return them right - adjusted with zero fill. If N exceeds the number of bytes remaining after positioning at byte I, or I or N are negative, the result is garbage.

MXPUTX (A,M,I) Store a byte (function or subroutine)

Take the right most byte from A and store it in byte I of word M. If I is greater than 10 or negative, the result is garbage.

CDC Specific Routines

BUFFER IN 9u,p) (a,b)

Transfers a physical record from unit u to memory beginning at word a, and ending at word b. p is a 1, indicating a 9-track labeled tape and no conversion is to be performed on the data.

BUFFER OUT 9u,p) (a,b)

Writes a physical record on unit u. Information to be written starts at word a, and ends at word b. p is a 1, indicating 7-track odd parity (binary) tape.

UNIT (u)

Function used to check status of unit u after a BUFFER IN or BUFFER OUT operation. Control returns to the program when the unit completes that operation. The function returns one of the following values:

-1.	unit ready, no end-of-file or parity errors.
+0.	unit ready, end-of-file encountered.
+1.	unit ready, parity error encountered.

LENGTHX (u, nw, ubc)

Subroutine called to retrieve information regarding the previous BUFFER IN operation on unit u. nw contains the number of 60 bit words read, and ubc contains the number of bits in the last word transferred.

Note: The CDC Fortran reference manual states:

"After an unformatted BUFFER IN on 9-track S or L tapes, the unused bit count parameter of LENGTHX is rounded down so as to indicate a whole number of 6-bit characters. For example, a BUFFER IN of a 23 character record returns a length of four words with an unused bit count of 54, even though the actual unused bit count is 56."

Hence, although this information (ubc) is readily available, the program does not use it because it is unreliable.

APPENDIX A

SAMPLE RECORD

SAMPLE RECORD

There is a sample record on the following page. The information below explains the structure of that particular record.

Leader

There are two important pieces of information in the leader, the LRECL and the BDA. In this case, the LRECL is 515 bytes, and the BDA is 145.

Record Directory

Entry 1 in the directory has a tag ØØ1, a field length of 13 bytes, and starting address Ø (relative to BDA).

Entry 2 has a tag ØØ8, a field length of 41 bytes, and starting address 13.

.

.

.

Entry 10 has a tag 65Ø, a field length of 59, and starting address 311.

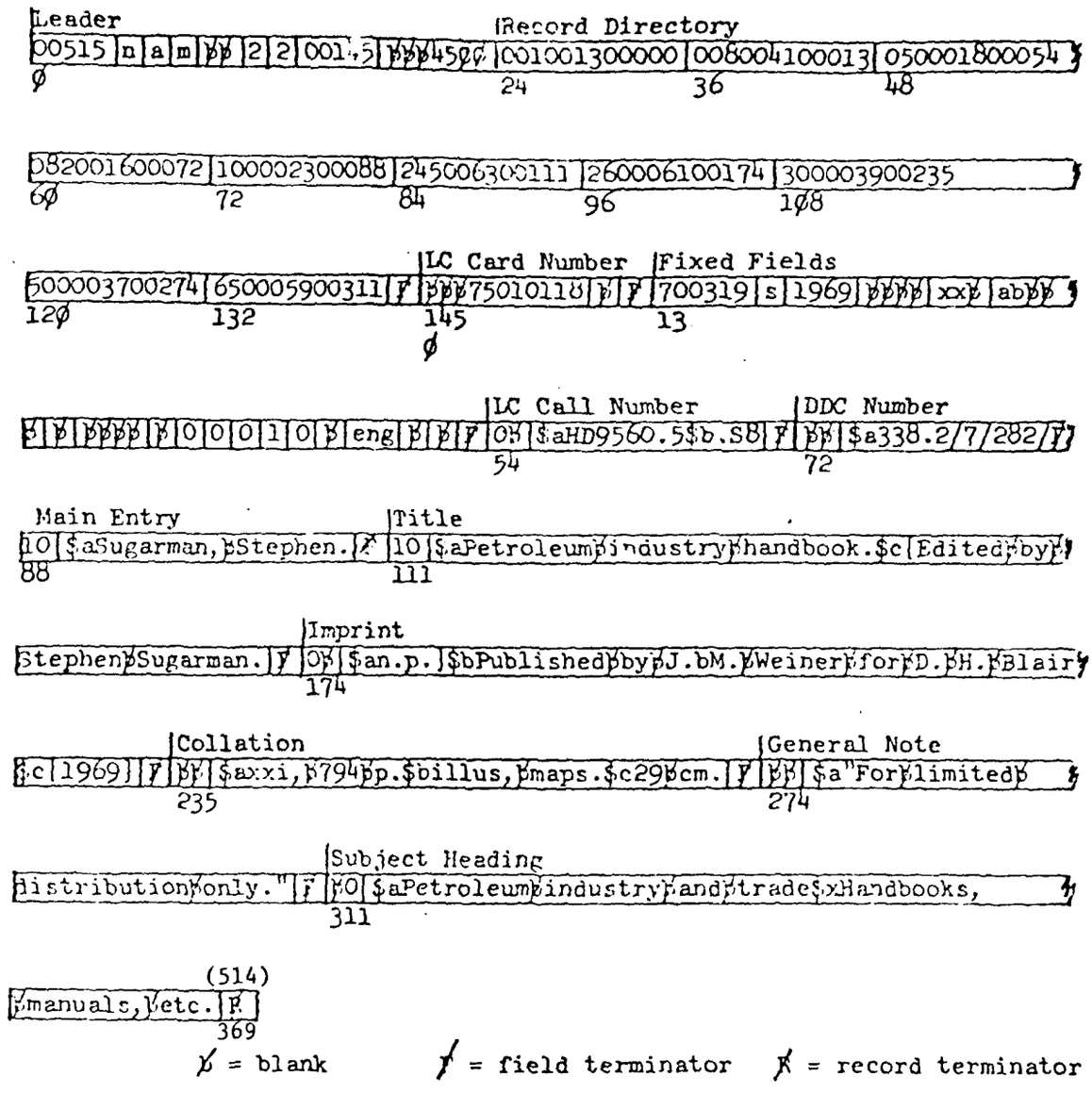
The directory ends with a field terminator.

Control/Variable Fields

The first field begins at address 145, the BDA. All control/variable fields starting addresses are determined relative to the BDA, which is the reason why the address for the first field is Ø. The last byte in the record has address 369 relative to the BDA, address 514 relative to the beginning of the record.

The record ends with a record terminator.

SAMPLE RECORD IN THE MARC FORMAT



A P P E N D I X B

8-BIT EXPANDED ASCII CHARACTER SET

EXPANDED 8-BIT ASCII CHARACTER SET

Dec.	Hex.	Binary	Graphic	Name and/or Function
Ø	ØØ	0000 0000		Null
1	Ø1	0000 0001		Start of Heading
2	Ø2	0000 0010		Start of Text
3	Ø3	0000 0010		End of Text
4	Ø4	0000 0100		End of Transmission
5	Ø5	0000 0101		Enquiry
6	Ø6	0000 0110		Acknowledge
7	Ø7	0000 0111		Bell
8	Ø8	0000 1000		Backspace
9	Ø9	0000 1001		Horizontal Tabulation
10	ØA	0000 1010		Line Feed
11	ØB	0000 1011		Vertical Tabulation
12	ØC	0000 1100		Form Feed
13	ØD	0000 1101		Carriage Return
14	ØE	0000 1110		Shift Out
15	ØF	0000 1111		Shift In
16	1Ø	0001 0000		Data Link Escape
17	11	0001 0001		Device Control 1
18	12	0001 0010		Device Control 2
19	13	0001 0011		Device Control 3
2Ø	14	0001 0100		Device Control 4
21	15	0001 0101		Negative Acknowledge
22	16	0001 0110		Synchronous Idle
23	17	0001 0111		End of Transmission Block
24	18	0001 1000		Cancel
25	19	0001 1001		End of Medium
26	1A	0001 1010		Substitute
27	1B	0001 1011		Escape
28	1C	0001 1100		End of File
29	1D	0001 1101		End of Record
30	1E	0001 1110		Field Terminator

Dec.	Hex.	Binary	Graphic	Name and/or Function
31	1F	0001 1111		Double Dagger (delimiter)
32	2Ø	0010 0000		Space
33	21	0010 0001	!	Exclamation Point
34	22	0010 0010	"	Quotation Marks
35	23	0010 0010	#	Number Sign
36	24	0010 0100	$	Dollar Sign
37	25	0010 0101	%	Percent Sign
38	26	0010 0110	&	Ampersand
39	27	0010 0111	'	Apostrophe
4Ø	28	0010 1000	(	Opening Parenthesis
41	29	0010 1001	)	Closing Parenthesis
42	2A	0010 1010	*	Asterisk
43	2B	0010 1011	+	Plus
44	2C	0010 1100	,	Comma
45	2D	0010 1101	-	Hyphen (minus)
46	2E	0010 1110	.	Period (decimal point)
47	2F	0010 1111	/	Slash
48	3Ø	0011 0000	Ø	
49	31	0011 0001	1	
5Ø	32	0011 0010	2	
51	33	0011 0011	3	
52	34	0011 0100	4	
53	35	0011 0101	5	
54	36	0011 0110	6	
55	37	0011 0111	7	
56	38	0011 1000	8	
57	39	0011 1001	9	
58	3A	0011 1010	:	Colon
59	3B	0011 1011	;	Semi-colon
6Ø	3C	0011 1100	<	Less Than

Dec.	Hex.	Binary	Graphic	Name and/or Function
61	3D	0011 1101	=	Equals
62	3E	0011 1110	>	Greater Than
63	3F	0011 1111	?	Question Mark
64	4Ø	0100 0000	@	Commercial At Sign
65	41	0100 0001	A	
66	42	0100 0010	B	
67	43	0100 0011	C	
68	44	0100 0100	D	
69	45	0100 0101	E	
7Ø	46	0100 0110	F	
71	47	0100 0111	G	
72	48	0100 1000	H	
73	49	0100 1001	I	
74	4A	0100 1010	J	
75	4B	0100 1011	K	
76	4C	0100 1100	L	
77	4D	0100 1101	M	
78	4E	0100 1110	N	
79	4F	0100 1111	O	
8Ø	5Ø	0101 0000	P	
81	51	0101 0001	Q	
82	52	0101 0010	R	
83	53	0101 0011	S	
84	54	0101 0100	T	
85	55	0101 0101	U	
86	56	0101 0110	V	
87	47	0101 0111	W	
88	58	0101 1000	X	
89	59	0101 1001	Y	
9Ø	5A	0101 1010	Z	

Dec.	Hex.	Binary	Graphic	Name and/or Function
91	5B	0101 1011	[	Opening Bracket
92	6C	0101 1100		Reverse Slash
93	5D	0101 1101	]	Closing Bracket
94	5E	0101 1110		
95	5F	0101 1111		
96	6Ø	0110 0000		
97	61	0110 0001	a	
98	62	0110 0010	b	
99	63	0110 0011	c	
1ØØ	64	0110 0100	d	
1Ø1	65	0110 0101	e	
1Ø2	66	0110 0110	f	
1Ø3	67	0110 0111	g	
1Ø4	68	0110 1000	h	
1Ø5	69	0110 1001	i	
1Ø6	6A	0110 1010	j	
1Ø7	6B	0110 1011	k	
1Ø8	6C	0110 1100	l	
1Ø9	6D	0110 1101	m	
11Ø	6E	0110 1110	n	
111	6F	0110 1111	o	
112	7Ø	0111 0000	p	
113	71	0111 0001	q	
114	72	0111 0010	r	
115	73	0111 0011	s	
116	74	0111 0100	t	
117	75	0111 0101	u	
118	76	0111 0110	v	
119	77	0111 0111	w	
12Ø	78	0111 1000	x	

Dec.	Hex.	Binary	Graphic	Name and/or Function
121	79	0111 1001	y	
122	7A	0111 1010	z	
123	7B	0111 1011		
124	7C	0111 1100		
125	7D	0111 1101		
126	7E	0111 1110		
127	7F	0111 1111		Delete
128	8Ø	1000 0000		
129	81	1000 0001		
13Ø	82	1000 0010		
131	83	1000 0011		
132	84	1000 0100		
133	85	1000 0101		
134	86	1000 0110		
135	87	1000 0111		
136	88	1000 1000		
137	89	1000 1001		
138	8A	1000 1010		
139	8B	1000 1011		
14Ø	8C	1000 1100		
141	8D	1000 1101		
142	8E	1000 1110		
143	8F	1000 1111		
144	9Ø	1001 0000		
145	91	1001 0001		
146	92	1001 0010		
147	93	1001 0011		
148	94	1001 0100		
149	95	1001 0101		
15Ø	96	1001 0110		
151	97	1001 0111		
152	98	1001 1000		
153	99	1001 1001		
154	9A	1001 1010		

Dec.	Hex.	Binary	Graphics	Name and/or Function
155	9B	1001 1011		
156	9C	1001 1101		
157	9D	1001 1101		
158	9E	1001 1110		
159	9F	1001 1111		
160	A0	1010 0000		
161	A1	1010 0001	Ł	Polish L - Uppercase
162	A2	1010 0010	Ø	Scandinavian 0 with Slash
163	A3	1010 0011	Đ	D with Crossbar-Uppercase
164	A4	1010 0100	Þ	Icelandic Thorn - Uppercase
165	A5	1010 0101	AE	
166	A6	1010 0110	OE	
167	A7	1010 0111	ʹ	Mi͡agkiĭ Znak
168	A8	1010 1000	·	Dot in Middle of Line
169	A9	1010 1001	♭	Musical Flat
170	AA	1010 1010	R	Subscript Patent Mark
171	AB	1010 1011	±	Plus or Minus
172	AC	1010 1100	ơ	
173	AD	1010 1101	ư	
174	AE	1010 1110	ʾ	Alif
175	AF	1010 1111		
176	B0	1011 0000	ʿ	ʿAyn
177	B1	1011 0001	ł	Polish l - Lowercase
178	B2	1011 0010	ø	Scandinavian o with Slash Lowercase
179	B3	1011 0011	đ	d with Crossbar - Lowercase
180	B4	1011 0100	þ	Icelandic Thorn - Lowercase
181	B5	1011 0101	ae	
182	B6	1011 0110	oi	
183	B7	1011 0111	ʺ	Tvërdyĭ Znak
184	B8	1011 1000	ı	Turkish i - Lowercase
185	B9	1011 1001	£	British Pound

Dec.	Hex.	Binary	Graphic	Name and/or Function
186	BA	1011 1010	ⅎ	Eth
187	BB	1011 1011		
188	BC	1011 1100	₀	
189	BD	1011 1101	ᵥ	
190	BE	1011 1110		
191	BF	1011 1111		
192	CØ	1100 0000		
193	C1	1100 0001		
194	C2	1100 0010		
195	C3	1100 0011		
196	C4	1100 0100		
197	C5	1100 0101		
198	C6	1100 0110		
199	C7	1100 0111		
2ØØ	C8	1100 1000		
2Ø1	C9	1100 1001		
2Ø2	CA	1100 1010		
2Ø3	CB	1100 1011		
2Ø4	CC	1100 1100		
2Ø5	CD	1100 1101		
2Ø6	CE	1100 1110		
2Ø7	CF	1100 1111		
2Ø8	DØ	1101 0000		
2Ø9	D1	1101 0001		
21Ø	D2	1101 0010		
211	D3	1101 0011		
212	D4	1101 0100		
213	D5	1101 0101		
214	D6	1101 0110		
215	D7	1101 0111		
216	D8	1101 1000		

Dec.	Hex.	Binary	Graphic	Name and/or Location
217	D9	1101 1001		
218	DA	1101 1010		
219	DB	1101 1011		
22Ø	DC	1101 1100		
221	DD	1101 1101		
222	DE	1101 1110		
223	DF	1101 1111		
224	EØ	0000	?	Pseudo Question
225	E1	1110 0001	`	Grave
226	E2	1110 0010	´	Acute
227	E3	1110 0011	^	Circumflex
228	E4	1110 0100	~	Tilde
229	E5	1110 0101	¯	Macron
23Ø	E6	1110 0110	˘	Breve
231	E7	1110 0111	˙	Superior Dot
232	E8	1110 1000	¨	Umlaut or Dieresis
233	E9	1110 1001	ˇ	Hacek
234	EA	1110 1010	°	Circle or Angstrom
235	EB	1110 1011	⁀	Ligature
236	EC	1110 1100	⁀	Ligature
237	ED	1110 1101	'	High Comma Diacritical
238	EE	1110 1110	˝	Couble Acute
239	EF	1110 1111	◡	Candrabindu
24Ø	FØ	1111 0000	¸	Cedilla
241	F1	1111 0001	˛	Right Hook
242	F2	1111 0010	.	Dot Below Character
243	F3	1111 0011	..	Double Dot Below Character
244	F4	1111 0100	°	Circle Below Character
245	F5	1111 0101	=	Double Underscore
246	F6	1111 0110	_	Underscore
247	F7	1111 0111	˛	Left Hook

Dec.	Hex.	Binary	Graphic	Name and/or Function
248	F8	1111 1000	¸	Right Cedilla
249	F9	1111 1001	◡	Upadhamānīya
250	FA	1111 1010	^	Double Tilde
251	FB	1111 1011	◡	Double Tilde
252	FC	1111 1100		
253	FD	1111 1101		
254	FE	1111 1110	·	High Comma (Centered)
255	FF	1111 1111		

GREEK, SUBSCRIPT, AND SUPERSCRIPT CHARACTERS
(8-BIT)

GREEK

The following Greek characters are preceded by the escape sequence ESCg($1B67_{16}$). This is a locking escape and must be ended by the escape sequence ESCs($1B73_{16}$) to return to the standard set.

Decimal	Oct.	Hex.	Binary	Graphic	Name
97	141	61	0110 0010	α	Alpha
98	142	62	0110 0010	β	Beta
99	143	63	0110 0011	Γ	Gamma

SUBSCRIPTS

The following subscript characters are preceded by the escape sequence ESCb ($1B62_{16}$). This is a locking escape and must be ended by the escape sequence ESCs ($1B73_{16}$) to return to the standard set.

Decimal	Oct.	Hex.	Binary	Graphic	Name
48		3Ø	0011 0000	Ø	Subscript Ø, etc.
49	61	31	0011 0001	1	
50	62	32	0011 0010	2	
51	63	33	0011 0011	3	
52	64	34	0011 0100	4	
53	65	35	0011 0101	5	
54	66	36	0011 0110	6	
55	67	37	0011 0111	7	
56	68	38	0011 1000	8	
57	69	39	0011 1001	9	

SUBSCRIPTS (CONTINUED)

Decimal	Oct.	Hex	Binary	Graphic	Name
4Ø	50	28	0010 1000	(	Open Parenthesis
41	51	29	0010 1001	)	Closed Parenthesis
43	52	2B	0010 1011	+	Plus
45	53	2D	0010 1101	-	Minus

SUPERSCRIPTS

Superscript characters have the same values as the subscript character set; however, the escape sequence to enter the superscript set is ESCp (1B7Ø). The escape sequence to return to the standard set is ESCs ($1B73_{16}$).

A P P E N D I X C

6-BIT CHARACTER SET

BLANK PAGE

42

STANDARD 6-BIT CHARACTER SET DERIVED FROM 8-BIT ASCII CHARACTER SET

Dec.	Octal	Binary	Table Position	Graphic	Name and/or Function
Ø	00	000 000	33		Space
1	01	000 001	34	!	Exclamation Point
2	02	000 010	34	"	Quotation Marks
3	03	000 011	36	#	Number Sign
4	04	000 100	37	$	Dollar Sign
5	05	000 101	38	%	Percent Sign
6	06	000 110	39	&	Ampersand
7	07	000 111	40	'	Apostrophe
8	10	001 000	41	(	Opening Parenthesis
9	11	001 001	42	)	Closing Parenthesis
10	12	001 010	43	*	Asterisk
11	13	001 011	44	+	Plus
12	14	001 100	45	,	Comma
13	15	001 101	46	-	Hyphen (minus)
14	16	001 110	47	.	Period (Decimal Point)
15	17	001 111	48	/	Slash
16	20	010 000	49	Ø	
17	21	010 001	50	1	
18	22	010 010	51	2	
19	23	010 011	52	3	
20	24	010 100	53	4	
21	25	010 101	54	5	
22	26	010 110	55	6	
23	27	010 111	56	7	
24	30	011 000	57	8	
25	31	011 001	58	9	
26	32	011 010	59	:	Colon
27	33	011 011	60	;	Semicolon
28	34	011 100	61	<	Less Than

Dec.	Octal	Binary	Table Position	Graphic	Name and/or Function
29	35	011 101	62	=	Equals
30	36	011 110	63	>	Greater Than
31	37	011 111	64	?	Question Mark
32	40	100 000			
33	41	100 001	98	a	
34	42	100 010	99	b	
35	43	100 011	100	c	
36	44	100 100	101	d	
37	45	100 101	102	e	
38	46	100 110	103	f	
39	47	100 111	104	g	
40	50	101 000	105	h	
41	51	101 001	106	i	
42	52	101 010	107	j	
43	53	101 011	108	k	
44	54	101 100	109	l	
45	55	101 101	110	m	
46	56	101 110	111	n	
47	57	101 111	112	o	
48	60	110 000	113	p	
49	61	110 001	114	q	
50	62	110 010	115	r	
51	63	110 011	116	s	
52	64	110 100	117	t	
53	65	110 101	118	u	
54	66	110 110	119	v	
55	67	110 111	120	w	
56	70	111 000	121	x	
57	71	111 001	122	y	
58	72	111 010	123	z	
59	73	111 011			Shift Code Nonstandard Set
60	74	111 100			

Dec.	Octal	Binary	Table Position	Graphic	Name and/or Function
61	75	111 101			Shift Code Nonstandard Set
62	76	111 110			Shift Code Nonstandard Set
63	77	111 111	128		Delete

6-BIT NONSTANDARD SET 1

Each character in this set is preceded by the shift character (73_8)

Dec.	Octal	Binary	Table Position	Graphic	Name and/or Function
0	00	000 000	1		Null
1	01	000 001	2		Start of Heading
2	02	000 010	3		Start of Text
3	03	000 011	4		End of Text
4	04	000 100	5		End of Transmission
5	05	000 101	6		Enquiry
6	06	000 110			Acknowledge
7	07	000 111	8		Bell
8	10	001 000	9		Backspace
9	11	001 001	10		Horizontal Tabulation
10	12	001 010	11		Line Feed
11	13	001 011	12		Vertical Tabulation
12	14	001 100	13		Form Feed
13	15	001 101	14		Carriage Return
14	16	001 110	15		Shift Out
15	17	001 111	16		Shift In
16	20	010 000	17		Data Link Escape
[illegible]	21	010 001	18		Device Control 1
18	22	010 010	19		Device Control 2
19	23	010 011	20		Device Control 3
20	24	010 100	21		Device Control 4
21	25	010 101	22		Negative Acknowledge
22	26	010 110	23		Synchronous Idle
23	27	010 111	24		End of Transmission Block
24	30	011 000	25		Cancel
25	31	011 001	26		End of Medium
26	32	011 010	27		Substitute
27	33	011 011	28		Escape

Dec.	Octal	Binary	Table Position	Graphic	Name and/or Function
28	34	011 100	29		End of File
29	35	011 101	30		End of Record
30	36	011 110	31		Field Terminator
31	37	011 111	32	‡	Double Dagger (delimiter)
32	40	100 000	65	@	Commercial At Sign
33	41	100 001	66	A	
34	42	100 010	67	B	
35	43	100 011	68	C	
36	44	100 100	69	D	
37	45	100 101	70	E	
38	46	100 110	71	F	
39	47	100 111	72	G	
40	50	101 000	73	H	
41	51	101 001	74	I	
42	52	101 010	75	J	
43	53	101 011	76	K	
44	54	101 100	77	L	
45	55	101 101	78	M	
46	56	101 110	79	N	
47	57	101 111	80	O	
48	60	110 000	81	P	
49	61	110 001	82	Q	
50	62	110 010	83	R	
51	63	110 011	84	S	
52	64	110 100	85	T	
53	65	110 101	86	U	
54	66	110 110	87	V	
55	67	110 111	88	W	
56	70	111 000	89	X	
57	71	111 001	90	Y	
58	72	111 010	91	Z	
59	73	111 011	92	[	Opening Bracket

Dec.	Octal	Binary	Table Position	Graphic	Name and/or Function
60	74	111 100	93	\	Reverse Slash
61	75	111 101	94	]	Closing Bracket
62	76	111 110			
63	77	111 111			

6-BIT NONSTANDARD SET II

Each character in this set is preceded by the shift character 75_8

Dec.	Octal	Binary	Table Position	Graphic	Name and/or Location
0	00	000 000			
1	01	000 001	162	Ł	Polish L - Uppercase
2	02	000 010	163	Ø	Scandinavin 0 with Slash
3	03	000 011	164	Đ	D with Cross Bar - Uppercase
4	04	000 100	165	Þ	Icelandic Thorn - Uppercase
5	05	000 101	166	Æ	
6	06	000 110	167	Œ	
7	07	000 111	168	ʹ	Miagkii Znak
8	10	001 000	169	·	Dot in Middle of Line
9	11	001 001	170	♭	Musical Flat
10	12	001 010	171	®	Subscript Patent Mark
11	13	001 011	172	±	Plus or Minus
12	14	001 100	173	ƒ	
13	15	001 101	174	℧	
14	16	001 110	175	ʼ	Alif
15	17	001 111	176		
16	20	010 000	177	ʿ	ʿAyn
17	21	010 001	178	ł	Polish l - Lowercase
18	22	010 010	179	ø	Scandinavin o with Slash - Lowercase
19	23	010 011	180	đ	D with Cross Bar - Lowercase
20	24	010 100	181	þ	Icelandic Thorn - Lowercase
21	25	010 101	182	æ	
22	26	010 110	183	œ	
23	27	010 111	184	ʺ	Tverdi Znak

Dec.	Octal	Binary	Table Position	Graphic	Name and/or Location
24	30	011 000	185	ı	Turkish i - Lowercase
25	31	011 001	186	£	British Pound
26	32	011 010	187	ð	Eth
28	34	011 100	189	ø	
29	35	011 101	190	œ	
30	36	011 110			
31	27	011 111			
32	40	100 000	225	?	Pseudo Question
33	41	100 001	226	`	Grave
34	42	100 010	227	´	Acute
35	43	100 011	228	^	Circumflex
36	44	100 100	229	~	Tilde
37	45	100 101	230	¯	Macron
38	46	100 110	231	˘	Breve
39	47	100 111	232	˙	Superior Dot
40	50	101 000	233	¨	Umlaut or Dieresis
41	51	101 001	234	ˇ	Hacek
42	52	101 010	235	°	Circle or Angstrom
43	53	101 011	236	⁀	Ligature
44	54	101 100	237	⁀	Ligature
45	55	101 101	238	,	High Comma Diacritical
46	56	101 110	239	˝	Double Acute
47	57	101 111	240	˘	Candrabidu
48	60	110 000	241	¸	Cedilla
49	61	110 001	242	˛	Right Hook
50	62	110 010	243	.	Dot Below Character
51	63	110 011	244	..	Double Dot Below Character
52	64	110 100	245	˚	Circle Below Character
53	65	110 101	246	‗	Double Underscore
54	66	110 110	247	_	Underscore
55	67	110 111	248	˩	Left Hook

Dec.	Octal	Binary	Table Position	Graphic	Name and/or Location
56	70	111 000	249	¢	Right Cedilla
57	71	111 011	250	◡	Upadhmaniya
58	72	111 010	251	~	Double Tilde
59	73	111 011	252	◡	Double Tilde
60	74	111 100			
61	75	111 100			
62	76	111 110	255	'	High Comma (Centered)
63	77	111 111			

GREEK, SUBSCRIPT, AND SUPERSCRIPT CHARACTERS (6-BIT)

GREEK

The following Greek characters are preceded by the escape sequence shift ESCg (733347_8). This is a locking sequence and must be ended by the escape sequency shift ESCs (733363_8) to return to the standard set.

Decimal	Octal	Binary	Graphic	Name
33	41	100 001	α	Alpha
34	42	100 010	β	Beta
35	43	100 011	Γ	Gamma

SUBSCRIPTS

The following subscript characters are preceded by the escape sequence shift ESCb (733342_8) and must be ended by the escape Shift ESCb (733363_8) to return to the standard set.

Decimal	Octal	Binary	Graphic	Name
16	20	010 000	0	Subscript 0, etc.
17	21	010 001	1	
18	22	010 010	2	
19	23	010 011	3	
20	24	010 100	4	
21	25	010 101	5	
22	26	010 110	6	
23	27	010 111	7	
24	30	010 000	8	
25	31	010 001	9	
8	10	001 000	(	Open Parenthesis
9	11	001 001	)	Closed Parenthesis
11	13	001 011	+	Plus
13	15	001 101	-	Minus

C. SUPERSCRIPTS

Superscript characters have the same values as the subscript set; however,the escape sequence to enter the superscript set is shift ESCp (733360_8). The escape sequence to return to the standard set is shift ESCs (733363_8).

REFERENCES

(1) Books: A MARC Format, U. S. Library of Congress, MARC Development Office, Fifth Edition, 1972.

AFGL Library
Z
699.4
M2
U535
1972


FILME
40